AMENDMENTS TO THE CLAIMS

1. (Original) A method for enhancing dynamic timing simulation comprising:

accessing a netlist comprising combinational logic nodes, including output nodes, interconnections, and input and output storage elements;

assigning a delay to each of said nodes;

determining a maximum forward delay sum for each node;

determining a safe delay period for each of said output nodes;

removing timing checks from those output nodes for which the maximum forward delay sum is less than the safe delay period;

determining a minimum reverse delay difference for each of a portion of said nodes;

identifying the nodes for which the minimum reverse delay difference is greater than the maximum forward delay sum;

setting the delays for the identified nodes to zero; and

performing dynamic timing simulation.

2. (Currently Amended) The method of claim 1, wherein the forward maximum delay sum includes an interconnect delay~~;~~.

3. (Original) The method of claim 1, wherein the delay assigned to at least one of said nodes is derived from a gate delay.

4. (Original) The method of claim 1, wherein the fundamental unit for deriving the node delays is a gate.

5. (Original) The method of claim 1, wherein at least one

of said output nodes is associated with a sequential element.

6. (Original) The method of claim 5, wherein said sequential element is a flip—flop.

7. (Original) The method of claim 6, further including the setting to zero of delays associated with clock circuit buffers driving said flip—flop.

8. (Original) The method of claim 5, wherein said safe delay period is derived from a clock period minus a setup time.

9. (Original) The method of claim 1, further including accessing a standard delay format (SDF) file to obtain delay information.

10. (Original) The method of claim 9, wherein said delay information relates to data dependent delays.

11. (Original) The method of claim 1, further including setting the clock-to—Q delay to zero for input storage elements that have had the delays of all connected nodes set to zero.

12. (Original) A computer readable medium containing executable instructions which, when executed in a processing system, causes the system to perform the steps for enhancing the runtime speed of a logic simulator, comprising:

defining a combinational portion of a logic circuit as a network comprising nodes, including output nodes, interconnections, and input and output storage elements;

assigning a delay to each of said nodes;

determining a maximum forward delay sum for each node;

determining a safe delay period for each of said output nodes;

removing timing checks from those output nodes for which the maximum forward delay sum is less than the safe delay period;

determining a minimum reverse delay difference for each of a portion of said nodes;

identifying the nodes for which the minimum reverse delay difference is greater than the maximum forward delay sum;

setting the delays for the identified nodes to zero; and

compiling the logic simulator.

13. (Currently Amended) The computer readable medium of claim 12, wherein the forward maximum delay sum includes an interconnect delay~~;~~.

14. (Original) The computer readable medium of claim 12, wherein the delay assigned to at least one of said nodes is a gate delay.

15. (Original) The computer readable medium of claim 12, wherein the fundamental unit for deriving the node delays is a gate.

16. (Original) The computer readable medium of claim 12, wherein at least one of said output nodes is associated with a storage element.

17. (Original) The computer readable medium of claim 16, wherein said storage element is a flip-flop.

18. (Currently Amended) The computer readable medium of

claim 17, further including instructions for ~~to~~ zero delays associated with clock circuit buffers driving said flip-flop.

19. (Original) The computer readable medium of claim 12, wherein said safe delay period is derived from a clock period and a setup time.

20. (Original) The computer readable medium of claim 19, further including instructions for accessing a standard delay format (SDF) file to obtain delay information.

21. (Currently Amended) The ~~method~~ computer readable medium of claim 20, wherein said delay information relates to data dependent delays.

22. (Original) The computer readable medium of claim 19, further including instructions for setting the clock-to-Q delay to zero for input storage elements that have had the delays of all connected nodes set to zero.

23. (Original) A system for enhancing the runtime speed of a logic simulator comprising a computer system, said computer system further comprising instructions for:

defining a combinational portion of a logic circuit as a network comprising nodes, including output nodes, interconnections, and input and output storage elements;

assigning a delay to each of said nodes;

determining a maximum forward delay sum for each node;

determining a safe delay period for each of said output nodes;

removing timing checks from those output nodes for which the maximum forward delay sum is less than the safe delay period;

determining a minimum reverse delay difference for each of said interior nodes and said input nodes;

identifying the nodes for which the minimum reverse delay difference is greater than the maximum forward delay sum;

setting the delays for the identified nodes to zero; and

compiling the logic simulator.

24. (Currently Amended) The system of claim 23, wherein the forward maximum delay sum includes an interconnect delay~~;~~.

25. (Original) The system of claim 23, wherein the delay assigned to at least one of said nodes is from a gate delay.

26. (Original) The system of claim 23, wherein the fundamental unit for deriving the node delays is a gate.

27. (Original) The system of claim 23, wherein at least one of said output nodes is associated with a storage element.

28. (Original) The system of claim 27, wherein said storage element is a flip—flop.

29. (Original) The system of claim 28, wherein said computer system comprises instructions for setting to zero of delays associated with clock circuit buffers driving said flip-flop.

30. (Original) The system of claim 23, wherein said safe delay period is derived from a clock period and a setup time.

31. (Original) The system of claim 23, wherein said

computer system comprises instructions for accessing a standard delay format (SDF) file to obtain delay information.

32. (Currently Amended) The ~~method~~ system of claim 31, wherein said delay information relates to data dependent delays.

33. (Original) A method of performing dynamic simulation comprising:

a) performing a delay assessment on a netlist comprising and sequential cells, said delay assessment assigning information for respective nodes in said netlist;

b) removing timing checks on sequential elements indicated as exempt from timing checks based on said delay information;

c) assigning zero delay to certain gates based on said delay information; and

d) performing dynamic simulation on said netlist, wherein said dynamic simulation enhances performance by:

performing cycle based simulation with respect to gates having assigned thereto zero delay as indicated by c); and

skipping timing checks for exempt sequential elements as indicated by b).

34. (Original) A method as described in Claim 33 wherein said delay information indicates a maximum delay at each node.

35. (Original) A method as described in Claim 33 wherein said delay assessment comprises:

a1) at each input node of said netlist, starting with a zero delay and traversing forward through each circuit path of said netlist; and

a2) during said traversing, aggregating maximum delays and assigning aggregated maximum delays to each node of each circuit path.

36. (Original) A method as described in Claim 33 wherein a) comprises determining exempt sequential elements by identifying output nodes each having a respective aggregated maximum delay that is less than a clock period minus a respective setup delay.

37. (Original) A method as described in Claim 36 wherein C) comprises:

for each circuit path terminating at an exempt sequential element, traversing backward through such circuit path to determine a partial circuit path for which zero delay can be designated to all gates therein.

38. (Original) A method as described in Claim 37 wherein said traversing backward comprises:

starting with said clock period minus said setup delay at an exempt sequential cell;

for each node traversed backward, subtracting a gate delay and assigning a lowest delay result value to said each node; and

continuing to traverse backward only if said value is greater than said node's aggregate maximum delay.

39. (Original) A system comprising a processor coupled to a bus and memory coupled to said bus wherein said memory contains instructions that when executed on said processor implements a method of performing dynamic simulation, said method comprising:

a) performing a delay assessment on a netlist comprising gates and sequential cells, said delay assessment assigning delay information for respective nodes in said netlist;

b) removing timing checks on sequential elements indicated as exempt from timing checks based on said delay information;

c) assigning zero delay to certain gates based on said delay information; and

d) performing dynamic simulation on said netlist, wherein said dynamic simulation enhances performance by:

performing cycle based simulation with respect to gates having assigned thereto zero delay as indicated by c); and

skipping timing checks for exempt sequential elements as indicated by b).

40. (Original) A system as described in Claim 39 wherein said delay information indicates a maximum delay at each node.

41. (Original) A system as described in Claim 39 wherein said delay assessment comprises:

a1) at each input node of said netlist, starting with a zero delay and traversing forward through each circuit path of said netlist; and

a2) during said traversing, aggregating maximum delays and assigning aggregated maximum delays to each node of each circuit path.

42. (Original) A system as described in Claim 39 wherein a) comprises determining exempt sequential elements by identifying output nodes each having a respective aggregated maximum delay that is less than a clock period minus a respective setup delay.

43. (Original) A system as described in Claim 42 wherein c) comprises:

for each circuit path terminating at an exempt sequential element, traversing backward through such circuit path to determine a partial circuit path for which zero delay can be designated to all gates therein.

44. (Original) A system as described in Claim 43 wherein said traversing backward comprises:

starting with said clock period minus said setup delay at an exempt sequential cell;

for each node traversed backward, subtracting a gate delay and assigning a lowest delay result value to said each node; and

continuing to traverse backward only if said value is greater than said node's aggregate maximum delay.